FMS–11/RSX
Mini–Reference

Order No. AV-H856A-TC

Operating Systems: RSX-11M Version 3.2
RSX-11M-PLUS Version 1.0

Use the following SDC order number to order a binder without the contents: AV-H780A-TK

To order additional copies of this document, contact the Software Distribution Center, Digital Equipment Corporation, Maynard, Massachusetts 01754

digital equipment corporation · maynard, massachusetts

First Printing, January 1980

The information in this document is subject to change without notice and should not be construed as a commitment by Digital Equipment Corporation. Digital Equipment Corporation assumes no responsibility for any errors that may appear in this document.

The software described in this document is furnished under a license and may only be used or copied in accordance with the terms of such license.

No responsibility is assumed for the use or reliability of software on equipment that is not supplied by DIGITAL or its affiliated companies.

Copyright © 1980 by Digital Equipment Corporation

The following are trademarks of Digital Equipment Corporation:

DIGITAL
DEC
PDP
DECUS
UNIBUS
COMPUTER LABS
COMTEX
DDT
DECCOMM
ASSIST-11
VAX
DECnet
DATATRIEVE
DECsystem-10
DECtape
DIBOL
EDUSYSTEM
FLIP CHIP
FOCAL
INDAC
LAB-8
DECSYSTEM-20
RTS-8
VMS
IAS
TRAX
MASSBUS
OMNIBUS
OS/8
PHA
RSTS
RSX
TYPESET-8
TYPESET-11
TMS-11
ITPS-10
SBI
PDT

7/81/14

CONTENTS

Page

FMS-11 FORM EDITOR (FED) 2

Starting the Form Editor 2

File Specifications and Options 2

The Form Editor Keypad 3

Using Form Editor Commands 4

Edit Status Information 4

Mode-Changing Functions 6

Cursor Movement Functions 7

Repeat Functions 7

Text Modification Functions 8

Assigning Video Attributes 10

Special Functions for Completing Questionnaires 10

Form-Wide Attributes Questionnaire 11

Field Attributes Questionnaire 13

Named Data Questionnaire 16

FMS-11 FORM UTILITY (FUT) 17

Starting the Form Utility 17

Form Utility Command Options 18

FMS-11 FORM DRIVER (FDV) 19

Form Driver Arguments for BASIC, COBOL, and FORTRAN 19

Form Driver Calls for BASIC, COBOL, and FORTRAN 20

Keywords for MACRO-11 Calls 24

MACRO-11 Calls and Function Codes 25

Status Codes Returned and Meanings 28

Sample MACRO-11 Call and Expansion 30

Necessary and Function-Dependent Arguments for MACRO-11 Calls 30

Necessary Arguments: Required Argument List Pointer (F$REQ) 31

Necessary Arguments: Function Code (F$FNC) 32

Global Symbols for Field Terminators 33
Field Terminator Codes 34
FMS-11 APPLICATIONS 34
Building BASIC-PLUS-2 Tasks 35
Building COBOL Tasks 36
Building FORTRAN Tasks 36
Building MACRO-11 Tasks 37

DOCUMENTATION CONVENTIONS

The following documentation conventions are used in this handbook:

Lowercase letters designate a variable whose value is determined when the command is entered.

Uppercase letters designate a command line element that must be entered as shown.

Square brackets ([ ]) enclose optional parameters, except for UIC specifications which always require brackets.

Numeric values in commands can be octal or decimal, unless the command description specifies otherwise.

Braces ({ }) indicate that the user can choose any one of the items that are enclosed.

xxx	A one- to three-character symbol indicates that you press a key on the terminal, for example, RET or ESC.
CTRL/n	The symbol CTRL/n indicates that you press the key labeled CTRL while you press another key, for example CTRL/C, CTRL/Y, CTRL/O.

FMS–11 FORM EDITOR (FED)

STARTING THE FORM EDITOR

If the Form Editor is installed on your system, you can start it by typing:

```
FED RET
```

or

```
FED command-line RET
```

If the Form Editor is not installed, you can start it by typing the MCR command:

```
MCR> RUN $FED RET
```

FILE SPECIFICATIONS AND OPTIONS

To create a new form, use the /CR option:

```
FED> /CR RET
```

To edit a form file or form library, use the filename:

```
FED> filename RET
```

To find the version number of the Form Editor that you are using, use the /ID option:

```
FED> /ID RET
```

For the RSX form file and library file specification use:

dev:[UIC] filename.type;version

Defaults:
device: system device

UIC:	current running UIC
input file type:	.FRM (form file)
input version #:	latest version

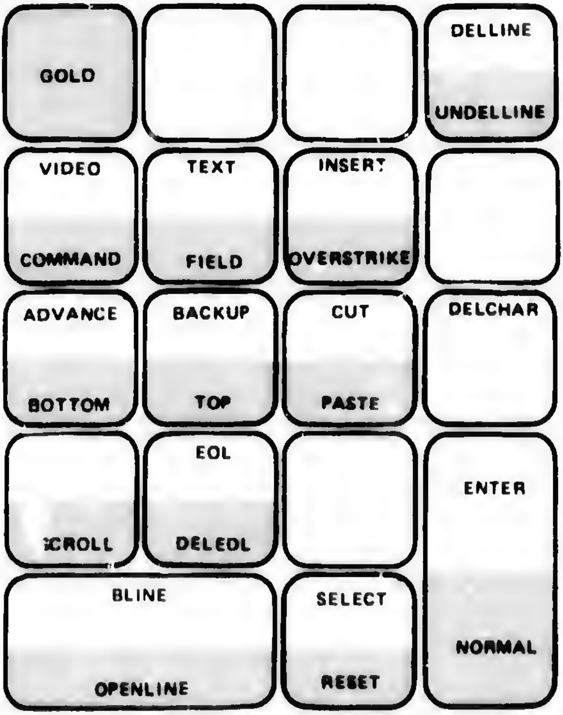
The output file type is .FRM. The output file name is the form name. The output version number is the highest version plus one.

THE FORM EDITOR KEYPAD

To use the FMS FORM EDITOR (FED), you need the keypad diagram to show you the form editing functions each keypad key controls.

To use the function on the upper part of the key, press that function key. To use the function on the lower part of the key, first press the GOLD (PF1) key and then press the function key.

FORM EDITOR KEYPAD LAYOUT



USING FORM EDITOR COMMANDS

To use a FED command at any time, press the GOLD key and then the COMMAND function key to get the COMMAND: prompt. In response to the FED COMMAND: prompt, type one of the commands listed below and then press the ENTER function key. Type the HELP command to display the Form Editor commands and their meanings.

Command	Meaning
EDIT	To create or edit a form image.
ASSIGN [option]	To assign field attributes, use with one of the following options:
ALL	To assign attributes for old and new fields.
NEW	To assign attributes for newly created fields only. N is the default ASSIGN command option.
FIELD fldnam	To assign attributes for the field only.
FORM	To assign form-wide attributes.
NAME	To enter and edit named data.
SAVE	To store the completed form and return to the system prompt.
QUIT	To cancel the current session and return to the system prompt.

To return to command level during the EDIT phase, press the GOLD key and then the COMMAND key on the keypad.

EDIT STATUS INFORMATION

When you are working in the EDIT phase, FED displays a status line with three categories of information about the CURSOR, about the current MODES, and about SELECT. The data about SELECT appears only after you use the SELECT function. Each category has several fields and the terms and abbreviations that appear in the fields are listed below.

Cursor

Field 1	TXT	Appears when the cursor is on a text character.
	FLD	Appears when the cursor is on a field picture character.
Field 2	NOR	Appears when the cursor is on a normal screen line.
	SCR	Appears when the cursor is on a line of a scrolled area.
Field 3	LIN	The number that follows is the cursor's screen line.
Field 4	COL	The number that follows is the cursor's screen column.

Modes

Field 1	TXT	Appears when you are editing in the text mode.
	FLD	Appears when you are editing in the field mode.
Field 2	ADV	Appears when you are editing in the advance mode.
	BCK	Appears when you are editing in the back-up mode.
Field 3	INS	Appears when you are editing in the insert mode.
	OVS	Appears when you are editing in the overstrike mode.

Select

Field 1	LIN	The number that follows is the select point's screen line.
Field 2	COL	The number that follows is the select point's screen column.

MODE-CHANGING FUNCTIONS

To change modes while using the Form Editor press the appropriate keys on the keypad. The mode functions are named on the key.

Keypad Key	Function
TEXT	In the TEXT mode, all characters entered become background text in the form.
FIELD	In the FIELD mode, only picture-validation characters and field-marker characters are accepted. They define the form's fields.
INSERT	In the INSERT mode, the character you type is inserted and shifts the cursor's character and the remaining characters on the line to the right.
OVERSTRIKE	In the OVERSTRIKE mode, FED replaces the cursor's character with the character you type and moves the cursor to the right.
ADVANCE	In the ADVANCE mode, the direction of the cursor movement for the keypad functions BLINE and EOL changes. The functions move the cursor to the right and down until the BACKUP function is used.
BACKUP	In the BACKUP mode, the direction of the cursor movement for the BLINE and EOL changes. The functions move the cursor to the left and up until the ADVANCE function is used.

FIELD-MARKER CHARACTERS		**PICTURE-VALIDATION CHARACTERS**	
!	(		
"	)	9	Numeric
#	*	A	Alphabetic
$	+	C	Alphanumeric
%	,	N	Signed numeric
&	-	X	Any ASCII
'	.		character from
/	=		040 octal
:	>		through 176
;	?		
<	@		

CURSOR MOVEMENT FUNCTIONS

Keyboard Key	Function
→	Moves the cursor to the next character position.
←	Moves the cursor to the preceding character position.
↑	Moves the cursor to the character position above.
↓	Moves the cursor to the character position below.
RET	Moves the cursor to the beginning of a line.

Keypad Key	Function
BLINE	Moves cursor to the beginning of a line.
EOL	Moves cursor to the end of a line.
TOP	Moves the cursor to the upper left corner of the screen.
BOTTOM	Moves the cursor to the lower right corner of the screen.

REPEAT FUNCTIONS

If the VT100 autorepeat feature is enabled, you can repeat any single-key function by holding down the key that controls the function.

REPEAT	To use the REPEAT function, press the GOLD key and type an integer. FED then repeats the next function (any function except DELETE) or inserts a single character the number of times you have specified.

TEXT MODIFICATION FUNCTIONS

To activate text modification functions, first type the keypad function key and then use the keyboard keys to insert the characters that are valid for the mode (TEXT or FIELD).

Keypad Key	Function
OPENLINE	Inserts a blank line by moving the cursor's current line and lines below it down. FED reassigns screen locations for all fields in the lines that are moved down.
DELETE	Erases the preceding character. To use the DELETE function, press the DELETE key. In the INSERT mode, the DELETE function deletes the character to the left of the cursor, moves everything to the left, and puts a space at the end of the line. In the OVERSTRIKE mode, the DELETE function deletes the character to the left of the cursor, replaces the character with a space, and moves the cursor to that position.
DELCHAR	Erases the cursor's character. In the INSERT mode, the DELCHAR function deletes the cursor's character, moves everything else one character to the left, and puts a space at the end of the line. In the OVERSTRIKE mode, the DELCHAR function replaces the cursor's character with a space and moves the cursor one character to the right.
DELLINE	Erases the cursor's line, stores the line in the line buffer, and moves up the lines below. FED reassigns screen locations for all fields in lines that are moved.

(continued on next page)

Keypad Key	Function
UNDELLINE	Inserts the contents of the line buffer at the cursor's line. FED reassigns screen locations for all fields in lines that are moved.
SELECT	Marks one corner of a rectangular select range for cutting or for assigning video attributes. When you move the cursor, the cursor marks the opposite corner of the select range.
CUT	Erases the select range, stores the erased string in the paste buffer, and replaces the block with spaces.
PASTE	Inserts the contents of the paste buffer if the target paste area has only spaces. If the paste area contains a non-blank character or if the contents of the paste buffer cannot fit within screen boundaries, paints the target paste area in reverse video and displays the message "Cannot paste over margins or non-blanks." When the PASTE function is not possible, press any key to display the paste area, and then move the cursor to define an acceptable paste area.
SCROLL	Defines all fields in the cursor's current line as scrolled fields. Because the Form Driver does not display constant text in a scrolled region, a scrolled region should contain only field-marker characters and picture validation characters. All lines in a scrolled region must be identical.
DELEOL	Erases to the end of the cursor's line and stores the erased string in the line buffer.
CTRL/U	Erases to the beginning of the cursor's line and stores the erased string in the line buffer.

ASSIGNING VIDEO ATTRIBUTES

To assign video attributes, create a select range and then use the VIDEO function on the keypad. When FED displays the VIDEO:prompt, type any of the following responses, and then press either the ENTER or RETURN key:

Attribute	Function
BOLD	Displays the characters in bold face.
BLINK	Displays the characters as blinking.
REVERSE	Displays the characters on a reversed background.
UNDERLINE	Underlines the characters.
CLEAR	Removes all video attributes from the characters.
EDIT	Stops assigning video attributes and returns to the normal screen editing mode.

SPECIAL FUNCTIONS FOR COMPLETING QUESTIONNAIRES

Use the following Form Driver functions while you are working with any of the Form Editor attribute assignment questionnaires.

Key	Function
HELP	Displays a one-line explanation of the current questionnaire entry.
TAB	Assigns the current attribute value and moves to the next entry.
BACKSPACE	Moves the cursor to the beginning of the preceding entry.
←	Moves the cursor to the preceding character within the current entry.

(continued on next page)

Key	Function
→	Moves the cursor to the next character within the current entry.
DEL	Erases the preceding character within the current entry.
ENTER	Assigns current attribute values to the entire questionnaire: begins the next questionnaire or returns to the COMMAND: prompt.
RET	Assigns current attribute values to the entire questionnaire: begins the next questionnaire or returns to the COMMAND: prompt.
SELECT	Assigns current attribute values and displays the COMMAND: prompt.

FORM-WIDE ATTRIBUTES QUESTIONNAIRE

After the FORM command, FED displays the following questionnaire. If no form-wide attributes have been assigned, FED uses the defaults shown here:

```
Form-Wide Attributes Questionnaire

     Form Name              :
     Help Form Name         :
     Reverse Screen         (Y,N)   N
     Current Screen         (Y,N)   N
     Wide Screen            (Y,N)   N
     Starting Line          (1,23)  1
     Ending Line            (1,23)  23

     Impure Area                bytes
     Form Size                  words
```

Complete the fields in the form-wide attributes questionnaire as follows:

Field	Procedure
Form Name	Required. Type in a name from 1 to 6 alphanumeric characters long. The Form Editor converts lower-case letters to uppercase.
Help Form Name	Optional. Type in a name from 1 to 6 alphanumeric characters long. The Form Editor converts lower-case letters to uppercase.
Reverse Screen	Type Y if the form is to be displayed on a light background. Type N if the form is to be displayed on a dark background.
Current Screen	Type Y to display the form in the current screen mode (80-column or 132-column width and screen background). Type N to reset the screen if necessary to conform to the display modes of the form. Do not specify a Y for both the Current Screen and Wide Screen questions.
Wide Screen	Type Y to display the form in 132-column mode, and specify N to the Current Screen question. Type N to display the form in 80-column mode.
Starting Line	Type a number from 1 to 23 to identify the screen line that is the top line to be cleared when the form is displayed.
Ending Line	Type a number from 1 to 23 to identify the screen line that is the bottom line to be cleared when the form is displayed.
Impure Area	The Form Editor reports the number of bytes required for the Form Driver impure area.
Form Size	The Form Editor reports the number of words in the form description.

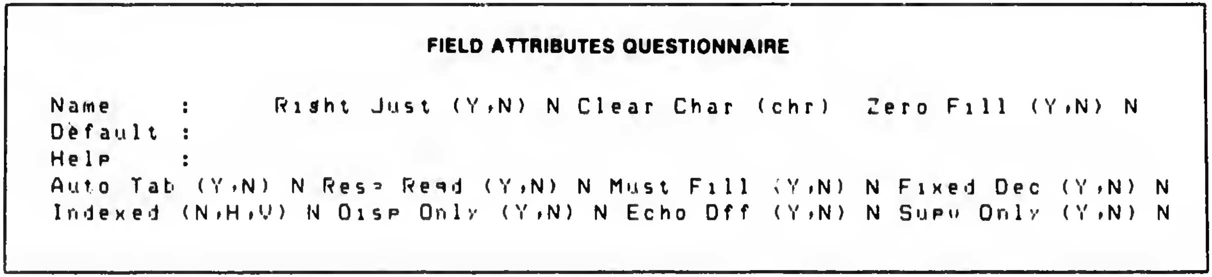
FIELD ATTRIBUTES QUESTIONNAIRE

After any ASSIGN command, the Form Editor displays the field attributes questionnaire shown on the following page.

When you type the ASSIGN NEW command, the questionnaire applies to the first field in the form to which attributes have not been assigned.

When you type the ASSIGN ALL command, the questionnaire applies to the first field in the form, whether or not attributes have been assigned to the field.

When you type the ASSIGN FIELD command, the questionnaire applies only to the field you name.



FIELD ATTRIBUTES QUESTIONNAIRE

```
Name     :      Right Just (Y,N) N Clear Char (chr)  Zero Fill (Y,N) N
Default  :
Help     :
Auto Tab (Y,N) N Resp Reqd (Y,N) N Must Fill (Y,N) N Fixed Dec (Y,N) N
Indexed (N,H,V) N Disp Only (Y,N) N Echo Off (Y,N) N Supv Only (Y,N) N
```

Complete the Field Attributes Questionnaire fields as follows:

Field	Procedure
Name	Type a name from 1 to 6 characters long. The name may be unique within the form. The first character must be alphanumeric. Later characters can be any characters in the range from ASCII octal 040 through 176.
Right Just	Type Y to right justify operator data within the field. Type N to left justify operator data within the field.
Clear Char	For a zero-filled field, type 0. For a field that is not zero-filled, type the character you want the Form Driver to use as a clear character. The default clear character is a space.
Zero Fill	Type Y if empty positions in the response are to be padded with zeroes. Type N if the operator response is to be padded with blanks.
Default	Optional. Type in the initial field value the Form Driver is to display. The default value cannot be longer than the field.
Help	Optional. Type in the message the Form Driver is to display when an operator presses the HELP key while working on the field.
Auto Tab	Type Y if the Form Driver is to move automatically to the next field when an operator enters the last character in the field. Type N if the operator is required to type an explicit field terminator.
Resp Reqd	Type Y if an operator must type an entry for the field. Type N if the field is optional for an operator.

(continued on next page)

Field	Procedure
Must Fill	Type Y if an operator must fill the field completely. Type N if a variable length response is acceptable.
Fixed Dec	Type Y if the field is numeric and contains a single decimal point as a field marker character. Type N for other fields.
Indexed	Type H or V if the field is indexed. When you type H, the Form Driver moves horizontally to the next horizontally indexed field. When you type V, the Form Driver moves vertically to the next entry. Type N if the field is not indexed. The attribute is not valid within scrolled regions.
Disp Only	Type Y if the Form Driver is to display a value in the field but an operator cannot enter a value. Type N if an operator can enter a value in the field.
Echo Off	Type Y if the Form Driver is not to echo what an operator enters in the field. Type N if the Form Driver is to echo what an operator types.
Supv Only	Type Y if the Form Driver, when in the Supervisor Only mode, is to prohibit an operator response. Type N if the Form Driver is to accept an operator response in the Supervisor Only mode, as well as at all other times.

NAMED DATA QUESTIONNAIRE

After the NAME command, the Form Editor displays the following questionnaire. Each form can have up to 16 named data items. The Form Editor displays all existing named data for the form.

Named Data Questionnaire

Name	Data
. . . .	. . . .

Complete the fields in the named data questionnaire as follows:

Field	Procedure
Name	Optional. Type a name from 1 to 6 characters long. You can use any non-control characters (characters in the range from ASCII octal 040 through 176). Names must be unique within a form if the Form Driver is to retrieve the data reliably by name.
Data	Type the data that the Form Driver is to associate with the name or line number. The maximum length for the data string is 60 characters (characters in the range from ASCII octal 040 through 176).

THE FMS-11 FORM UTILITY (FUT)

STARTING THE FORM UTILITY

If FUT is installed on your system, there are two formats for calling the Form Utility:

```
FUT RET
```

or

```
FUT[[file-specification string][/options]] RET
```

The MCR RUN command format is:

```
RUN $FUT RET
```

FORM UTILITY COMMAND OPTIONS

To use FUT options, use this format:

```
FUT[file-specification-String]/option RET
```

(The only options that do not require file-specifications are /HE and /ID.)

The FUT options are as follows.

For control and help

/HE	Displays help text
/ID	Displays identification message
/LI	Lists library directory
/SP, /-SP	Spools listing to line printer

For creating form library files

/BA, /-BA	Block-aligns forms in library files
/CR	Creates library from libraries and forms
/DE	Deletes forms from library
/EX	Extracts forms to build library
/RP	Replaces forms in library

For processing and converting form descriptions

/CC	Creates a COBOL form description
/FD	Writes a form description
/FF	Creates a form file from a library form
/OB	Writes object module of forms

THE FMS–11 FORM DRIVER (FDV)

In this section, the names of the languages are shortened to:

BASIC means BASIC-PLUS-2
COBOL means COBOL-11
FORTRAN means FORTRAN IV and FORTRAN IV-PLUS
MACRO means MACRO-11

FORM DRIVER ARGUMENTS FOR BASIC, COBOL, AND FORTRAN

CHAN	channel number An integer specifying a form library channel number (LUN).
FID	field identifier A string of up to 6 characters.
FIDX	field index An integer index into an array or into a named data list.
FLEN	field length An integer specifying the length of a field.
FLNM	file name A string that is a full or partial form library file specification.
FNAME	form name A string up to 6 characters.
FVAL	field value A string that is the input or output value for a field or is a named data value.
IMPURE	impure area
LINE	line number An integer that specifies the first line on the screen to be cleared when the form is displayed.
STATUS	return status An integer status code returned when the Form Driver completes a call.

STAT2	system status code A secondary status code that is a numeric value returned by the Form Driver when the status is FE$IOL (-4) or FE$IOR (-18).
TERM	terminator An integer code for the key that terminated input in a field.

FORM DRIVER CALLS FOR BASIC, COBOL, AND FORTRAN

FCLRSH
Clear entire screen and display form

BASIC	FCLRSH (fnam[,line])
COBOL	"FCLRSH" USING fnam[,line].
FORTRAN	FCLRSH (fnam[,line])

FGCF
Get current field name

BASIC	FGCF (fid[,fidx])
COBOL	"FGCF" USING fid[,fidx].
FORTRAN	FGCF (fid[,fidx])

FGET
Get specified field

BASIC	FGET (fval,term,tid[,fidx]) or FGET
COBOL	"FGET" USING fval,term,fid[,fidx]. or "FGET"
FORTRAN	FGET (fval,term,fid[,fidx]) or FGET

FGETAF
Get any field

BASIC	FGETAF (fval,term,fid[,fidx])
COBOL	"FGETAF" USING fval,term,fid[,fidx].
FORTRAN	FGETAF (fval,term,fid[,fidx])

FGETAL
Get all fields

BASIC	FGETAL ([fval[,term]])
	or
	FGETAL
COBOL	"FGETAL" USING [fval[,term]].
	or
	"FGETAL"
FORTRAN	FGETAL ([fval[,term]])
	or
	FGETAL

FIDATA
Get named data by index

BASIC	FIDATA (fidx,fval)
COBOL	"FIDATA" USING fidx,fval.
FORTRAN	FIDATA (fidx,fval)

FINIT
Initialize impure area

BASIC	FINIT (impure,size[,status])
COBOL	"FINIT" USING impure,size[,status].
FORTRAN	FINIT (impure,size[,status])

FINLN
Get current line of scrolled data

BASIC	FINLN (fid,fval[,term])
COBOL	"FINLN" USING fid,fval[,term].
FORTRAN	FINLN (fid,fval[,term])

FLCHAN
Set channel (LUN) for form library

BASIC	FLCHAN (chan)
COBOL	"FLCHAN" USING chan.
FORTRAN	FLCHAN (chan)

FLCLOS
Close form library file

BASIC	FLCLOS
COBOL	"FLCLOS"
FORTRAN	FLCLOS

FLEN
Return the length of the specified field

BASIC	FLEN (flen,fid[,fidx])
COBOL	"FLEN" USING flen,fid[,fidx].
FORTRAN	FLEN (flen,fid[,fidx])

FLOPEN
Open form library file

BASIC	FLOPEN (flnm)
COBOL	"FLOPEN" USING flnm
FORTRAN	FLOPEN (flnm)

FNDATA
Get named data by name

BASIC	FNDATA (fid,fval)
COBOL	"FNDATA" USING fid,fval.
FORTRAN	FNDATA (fid,fval)

FOUTLN
Output data to current line of scrolled area

BASIC	FOUTLN (fid[,fval])
COBOL	"FOUTLN" USING fid[,fval].
FORTRAN	FOUTLN (fid[,fval])

FPFT
Process the field terminator

BASIC	FPFT (term[,fid[,fval]])
	or
	FPFT
COBOL	"FPFT" USING term[fid[,fval]].
	or
	"FPFT"
FORTRAN	FPFT (term[,fid[,fval]])
	or
	FPFT

FPUT
Output data to specified field

BASIC	FPUT (fval,fid[,fidx])
COBOL	"FPUT" USING fval,fid[,fidx]
FORTRAN	FPUT (fval,fid[,fidx])

FPUTAL

Output the specified set of values for all fields, or if no values are specified, restore the default values to all fields in the form.

BASIC	FPUTAL (fval)
	or
	FPUTAL
COBOL	"FPUTAL" USING fval.
	or
	"FPUTAL"
FORTRAN	FPUTAL (fval)
	or
	FPUTAL

FPUTL

Output the specified value to the bottom screen line, or if no value is specified, clear the last line of the screen.

BASIC	FPUTL (fval)
	or
	FPUTL
COBOL	"FPUTL" USING fval.
	or
	"FPUTL"
FORTRAN	FPUTL (fval)
	or
	FPUTL

FRETAL

Return all fields

BASIC	FRETAL (fval)
COBOL	"FRETAL" USING fval.
FORTRAN	FRETAL (fval)

FRETN

Return the specified field

BASIC	FRETN (fval,fid[,fidx])
COBOL	"FRETN" USING fval,fid[,fidx].
FORTRAN	FRETN (fval,fid[,fidx])

FSHOW

Display specified form

BASIC	FSHOW (fnam[,line])
COBOL	"FSHOW" USING fnam[,line].
FORTRAN	FSHOW (fnam[,line])

FSPOFF
Turn supervisor-only mode off

BASIC	FSPOFF
COBOL	"FSPOFF"
FORTRAN	FSPOFF

FSPON
Turn supervisor-only mode on

BASIC	FSPON
COBOL	"FSPON"
FORTRAN	FSPON

FSTAT
Return the status codes for the last call

BASIC	FSTAT (stat[,stat2])
COBOL	"FSTAT" USING stat[,stat2].
FORTRAN	FSTAT (stat[,stat2])

KEYWORDS FOR MACRO-11 CALLS

Keyword	Meaning
ARG	Pointer to the main argument list.
FNC	The letters that follow the dollar sign ($) in each MACRO-11 function code.
LEN	Data length (in bytes). If an ASCIZ string is passed to the Form Driver, the length must be -1.
NAM	Pointer to an ASCIZ form library file specification or pointer to a 6-byte ASCII form name, field name, or data name.
NUM	Starting line number to display a form or the index value for a field or named data.
REQ	Pointer to the required arguments list.
TRM	Field terminator code returned by or passed to the Form Driver.
VAL	Pointer to data to be output by the Form Driver or data returned by the Form Driver.

MACRO-11 CALLS AND FUNCTION CODES

The calls are listed alphabetically by function codes. Formats are standardized for the MACRO-11 keyword calling convention. Each keyword is capitalized, and lower-case letters stand for references you must supply.

FC$ALL

Get the responses for all fields.

$FDV ARG=*a*,FNC=ALL,REQ=*b*

Outputs:

R0→	Argument list.
F$LEN(R0)	Total length of fields
F$TRM(R0)	Field terminator code.
F$VAL(R0)	Pointer to all responses.

FC$ANY

Get the response for any field.

$FDV ARG=*a*,FNC=ANY,REQ=*b*

Outputs:

R0→	Argument list.
F$LEN(R0)	Length of field.
F$NAM(R0)	Pointer to field name.
F$NUM(R0)	Field index.
F$TRM(R0)	Field terminator code.
F$VAL(R0)	Pointer to response.

FC$CLS

Close a form library.

$FDV ARG=*a*,FNC=CLS,REQ=*b*

Output:

R0→	Argument list.

FC$CSH

Clear the entire screen and show a form.

$FDV ARG=*a*,FNC=CSH,REQ=*b*,NAM=*c*,NUM=*d*

Output:

R0→	Argument list.

FC$DAT

Get named data.

$FDV ARG=*a*,FNC=DAT,REQ=*b*,NAM=*c*,NUM=*d*

Outputs:

R0→	Argument list.
F$LEN(R0)	Length of named data.
F$NAM(R0)	Pointer to data name.
F$VAL(R0)	Pointer to named data.

FC$GET

Get the response for a specific field.

$FDV ARG=*a*,FNC=GET,REQ=*b*,NAM=*c*,NUM=*d*

Outputs:

R0→	Argument list.
F$LEN(R0)	Length of field.
F$NAM(R0)	Pointer to field name.
F$NUM(R0)	Field index.
F$TRM(R0)	Field terminator code.
F$VAL(R0)	Pointer to response.

FC$GSC

Get the current line of a scrolled area.

$FDV ARG=*a*,FNC=GSC,REQ=*b*,NAM=*c*

Outputs:

R0→	Argument list.
F$LEN(R0)	Total length of fields in line.
F$TRM(R0)	Field terminator.
F$VAL(R0)	Pointer to responses for line

FC$LST

Output to the bottom line of the screen.

$FDV ARG=*a*,FNC=LST,REQ=*b*,VAL=*c*,LEN=*d*

Output:

R0→	Argument list.

FC$OPN

Open a form library.

$FDV ARG=*a*,FNC=OPN,REQ=*b*,NAM=*c*

Output:

R0→	Argument list.

FC$PAL

Output data to all fields.

$FDV ARG=*a*,FNC=PAL,REQ=*b*,VAL=*c*,LEN=*d*

Output:

R0→	Argument list.

FC$PSC

Output data to current line of scrolled area.

$FDV ARG=*a*,FNC=PSC,REQ=*b*,NAM=*c*,VAL=*d*,LEN=*e*

Output:

R0→	Argument list.

FC$PUT

Output data to a specific field.

$FDV ARG=*a*,FNC=PUT,REQ=*b*,NAM=*c*,NUM=*d*,
  VAL=*e*,LEN=*f*

Output:

R0→	Argument list.

FC$RAL

Return the responses for all fields.

```
$FDV ARG=a,FNC=RAL,REQ=b
```

Outputs:

R0→	Argument list.
F$LEN(R0)	Total length of fields.
F$VAL(R0)	Pointer to all responses.

FC$RTN

Return the response for a specific field.

```
$FDV ARG=a,FNC=RTN,REQ=b,NAM=c,NUM=d
```

Outputs:

R0→	Argument list.
F$LEN(R0)	Length of field.
F$VAL(R0)	Pointer to response.

FC$SHO

Show a form.

```
$FDV ARG=a,FNC=SHO,REQ=b,NAM=c,NUM=d
```

Output:

R0→	Argument list.

FC$SPF

Turn supervisor-only mode off.

```
$FDV ARG=a,FNC=SPF,REQ=b
```

Output:

R0→	Argument list.

FC$SPN

Turn supervisor-only mode on.

```
$FDV ARG=a,FNC=SPN,REQ=b
```

Output:

R0→	Argument list.

FC$TRM

Process field terminator.

```
$FDV ARG=a,FNC=TRM,REQ=b,TRM=c,NAM=d,
  VAL=e,LEN=f
```

Outputs:

R0→	Argument list.
F$NAM(R0)	Pointer to field name.
F$NUM(R0)	Field index.

STATUS CODES RETURNED AND MEANINGS

If the status code value returned by the Form Driver is:

>0	Form Driver completed call successfully
<0	An error was detected

The following status codes are returned to the calling program.

Status Code Returned	Status Code Value	Meaning
FS$SUC	1	Successful completion.
FS$INC	2	Current form incomplete.
FE$FCD	-1	Specified function code undefined.
FE$IMP	-2	Impure area too small.
FE$FSP	-3	Invalid file specification.
FE$IOL	-4	Error encountered opening form library. (The actual error code is returned in the second word of the status block or in STAT2.)
FE$FLB	-5	Specified file not form library.
FE$ICH	-6	Invalid channel number specified.
FE$FCH	-7	Form library not open on specified channel.
FE$FRM	-8	Invalid form definition.
FE$FNM	-9	Specified form does not exist.

(continued on next page)

Status Code Returned	Status Code Value	Meaning
FE$LIN	-10	Invalid first line number to display form.
FE$FLD	-11	Specified field does not exist.
FE$NOF	-12	No fields defined for current form.
FE$DSP	-13	Get call illegal for display only field(s).
FE$NSC	-14	Specified field is not in a scrolled area.
FE$DNM	-15	Named data specified does not exist.
FE$DLN	-16	Data specified to output too long (truncated by FDV).
FE$UTR	-17	Undefined field terminator.
FE$IOR	-18	Error encountered reading form library. (The actual error code is returned in the second word of the status block or in STAT2.)
FE$IFN	-19	Specified call invalid in current context of form.
	-20	Wrong number of arguments for FORTRAN call.
	-21	Impure area not initialized for a FORTRAN call.
FE$STR	-22	Insufficient room for string (BASIC-PLUS-2 only).

SAMPLE MACRO-11 CALL AND EXPANSION

F$ASIZ is the size (in bytes) of the main argument list and F$RSIZ is the size (in bytes) of the required argument list. The macro $FDVDF must be specified in a .MCALL statement and invoked as shown below to define these symbols so they can be used to allocate space for the argument lists.

```
                .MCALL          $FDV,$FDVDF
                $FDVDF

ARGLST:         .BLKB           F$ASIZ
REQLST:         .BLKB           F$RSIZ
```

A typical keyword-encoded call and the MACRO-11 expansion for it follow. ARGLST contains the main argument list. MACRO-11 generates the complete function code (FC$PUT) from the value supplied to FNC. REQLST contains the required argument list.

BUFFER contains the data, and DATLN contains the data length in bytes. FLD1 contains the field name, and the field index is 1.

```
$FDV ARGLST=#ARGLST,FNC=PUT,REQ=#REQLST,
  VAL=#BUFFER,LEN=DATLN,NAM=#FLD1,NUM=#1
```

Expansion:

```
                MOV             #ARGLST,R0
                MOV             #FC$PUT,F$FNC(R0)
                MOV             #REQLST,F$REQ(R0)
                MOV             #FLD1,F$NAM(R0)
                MOV             #1,F$NUM(R0)
                MOV             #BUFFER,F$VAL(R0)
                MOV             DATLN,F$LEN(R0)
                CALL            $FDV
```

NECESSARY AND FUNCTION-DEPENDENT ARGUMENTS FOR MACRO-11 CALLS

With MACRO-11 you can call the Form Driver with the instruction:

```
                JSR PC,$FDV     ;$FDV is global
                                ;R0 points to the following argument
                                 list:
```

(F$ASIZ is the size in bytes of the argument list.)

Offset	Meaning
F$FNC(R0)	One-word function code
F$REQ(R0)	One-word pointer to the required argument list
F$NAM(R0)	One-word pointer to an ASCIZ form library file specification or a 6-byte ASCII form name, field name, or data name
F$NUM(R0)	One-word starting line number to display a form or index value for field or named data
F$TRM(R0)	One-word field terminator code
F$VAL(R0)	One-word data pointer
F$LEN(R0)	One-word data length in bytes

NECESSARY ARGUMENTS: REQUIRED ARGUMENTS LIST POINTER (F$REQ)

F$REQ points to a required argument list with the following offsets:

Offset	Meaning
F$STS	one-word status block pointer
F$CHN	one-word channel number (LUN) for form I/O
F$IMP	one-word pointer to the impure area provided for the Form Driver

(F$RSIZ is the size in bytes of the required arguments list.)

NECESSARY ARGUMENTS: Function Code (F$FNC)

The following global symbols define Form Driver functions. In your application, you should always refer to the functions by the specified symbols, in order to ensure compatibility with future versions of the software.

Function Code	Meaning
FC$ALL	Get the responses for all fields
FC$ANY	Get the response for any field input
FC$CLS	Close form library
FC$CSH	Clear the entire screen and show the specified form
FC$DAT	Get named data
FC$GET	Get the response for the specified field
FC$GSC	Get the current line of a scrolled area
FC$LST	Output data to last line of display
FC$OPN	Open form library
FC$PAL	Output data to all fields
FC$PSC	Output data to the current line of a scrolled area
FC$PUT	Output data to a specified field
FC$RAL	Return the contents of all fields
FC$RTN	Return the contents of the specified field
FC$SHO	Show the specified form
FC$SPF	Turn supervisor-only mode off
FC$SPN	Turn supervisor-only mode on
FC$TRM	Process field terminator

GLOBAL SYMBOLS FOR FIELD TERMINATORS

The Form Driver detects the way an operator has terminated a field in a form or a form as a whole. Each terminator returns a different code. (The meanings for FORTRAN IV and BASIC are contained in parentheses.)

Code	Value	Meaning
FT$NTR	0	The RETURN key, or the keypad ENTER key.
FT$NXT	1	The TAB key (next field).
FT$PRV	2	The BACKSPACE key (previous field).
FT$ATB	3	Auto-tab action.
FT$XBK	4	The PF3 key (exit scrolled area backward).
FT$XFW	5	The PF4 key (exit scrolled area forward).
FT$SNX	6	Scroll forward to the next field.
FT$SPR	7	Scroll backward to the previous field.
FT$SFW	8	The downarrow key (scroll forward).
FT$SBK	9	The uparrow key (scroll backward).

FIELD TERMINATOR CODES

When the keypad has been set to application mode by the program, these are the codes:

Keypad Key	Numeric Code Returned
0	112
1	113
2	114
3	115
4	116
5	117
6	118
7	119
8	120
9	121
–	109
,	108
.	110

FMS–11 APPLICATIONS

To run the Form Driver configuration dialogue, install the FMS software. With it installed, enter the following command:

```
MCR> @[30,10]FDVBLD RET
```

BUILDING BASIC PLUS 2 TASKS

Building BASIC+2 tasks involves editing the CMD and ODL files produced by BASIC+2.

```
;
; BASDEM.CMD
;
;
;          Command file to build BASIC+2 demo
;
;
;          Underlined items are changes from file built with
;          BASIC+2 BUILD command.
;
BASDEM/CP,BASDEM/-SP=BASDEM/MP
;                     -----------     Add map file if desired.
UNITS = 14
ASG = TI:13:5
;            --                       Assign LUN 5 as terminal for
FDV.
ASG = SY:6:7:8:9:10:11:12
;         -                           Remove LUN 5 from this line.
//
;
```

```
;
; BASDEM.ODL
;
;
;          TKB Overlay Description File for BASIC+2 demo
;
;
;          Underlined items are changes from file built by
;          BASIC+2 BUILD command.
;
          .ROOT BIROT1-USER,RMS
USER:     .FCTR BASDEM-LIBR-FORM
;                               -----      Add FMS factor to root
LIBR:     .FCTR LB:[1,1]BASIC2/LB
;
FORM:     .FCTR HLLBP2-FRM1
;----------------------------              Add HLL interface
FRM1:     .FCTR FDVLRM/LB-LB:[1,1]RMSLIB/LB:RMSSYM
;------------------------------------------ Add FDV and define
;                                          RMS symbols
RMS:      .FCTR BIO017
@LB:[1,1]BASIC1
          .END
;
```

BUILDING COBOL TASKS

The following command file illustrates building a nonoverlaid COBOL task.

```
;
; CBLDEM.CMD
;
;
;       Command file to build COBOL demo
;
;
CBLDEM,CBLDEM/-SP=CBLDEM
;
;       COBOL interface and Form Driver library
;
HLLCBL,FDVLRM/LB
;
LB:[1,1]COBLIB/LB,RMSLIB/LB
/
UNITS=9
;       LUN 1 will be the terminal.
ASG = TI:1
;       LUN 3 will be the output file unit.
ASG = SY:3
;       Reassign the logical unit in the Form Driver.
GBLDEF = T$LUN:1
//
;
```

BUILDING FORTRAN TASKS

The following command files illustrate task build of FORTRAN tasks

```
;
; FORDEM.CMD
;
;
;       Build the FORTRAN demo
;
;
FORDEM,FORDEM/CR/-SP=FORDEM
;
;       FORTRAN interface and Form Driver library
;
HLLFOR,FDVLIB/LB
;
LB:[1,1]FOROTS/LB
//
;


;
; F4PFCS.CMD
;
;
```

```
;         Build the FORTRAN demo for FORTRAN IV+ using FCS
;
;
F4PDEM/FP,F4PDEM/CR/-SP=F4PDEM
;
;         FORTRAN interface and Form Driver library (using FCS)
;
HLLFOR,FDVLIB/LB
;
LB:[1,1]F4POTS/LB
//
;


;
; F4PRMS.CMD
;
;
;         Build the FORTRAN demo for FORTRAN IV+ using RMS
;
;
F4PDEM/FP,F4PDEM/CR/-SP=F4PDEM
;
;         FORTRAN interface and Form Driver library (using RMS)
;
HLLFOR,FDVLRM/LB
;
LB:[1,1]F4POTS/LB,LB:[1,1]RMSLIB/LB
//
;
```

BUILDING MACRO-11 TASKS

When assembling MACRO tasks, include the filespec FMSMAC/ML in the command line to resolve MACRO definitions for $FDV and other MACROS which you may want to use from this library.

The following TKB command file illustrates task building a MACRO task:

```
;
; MACTKB.CMD
;
;
;         Task Build the MACRO version of the DEMO application
;
;
MACDEM,MACDEM/-SP=MACDEM,FDVLIB/LB
/
UNITS=7
MAXBUF=512
//
;
```